JICAMARCA RAW and PROCESS DATA STRUCTURE

1. Working with the Header

Figure 1 shows the structure of a raw data file. The structure begins with the “FIRST HEADER” which contains the complete experimental information (e.g. Processing Characteristics), The second part is a block of data. After that the smaller header “BASIC HEADER” is saved followed by a block of data. From that point a smaller header (“BASIC HEADER”) precedes the block of data until the end of the raw file.

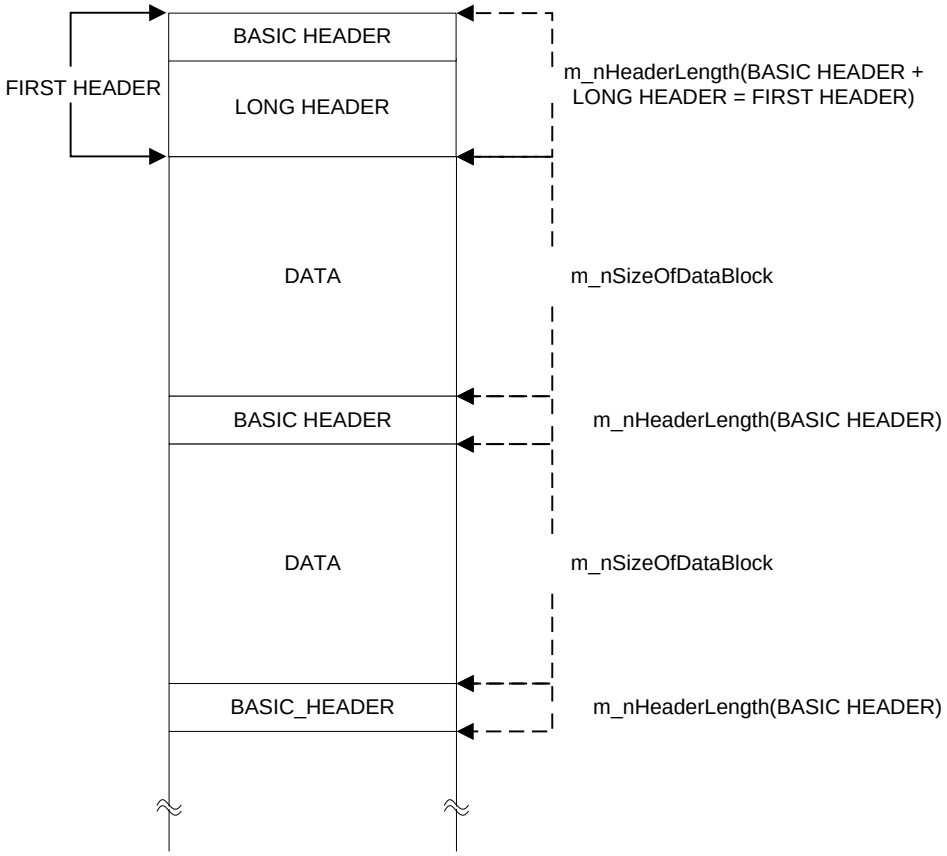

Figure 1 – Structure of Data File

1.1 Description of structures of the data file.

1.1.1 First Header

The "FIRST HEADER" contains the temporal information, radar controller and processing variables used in the experiment in order to obtain the raw data files. Basically we can divide the "FIRST HEADER" in two parts: A "BASIC HEADER" and "LONG HEADER". The "BASIC HEADER" will always have the temporal information of the data block that follows to the header and the "LONG HEADER" will have the radar and processing parameters.

1.1.2 Basic Header

The Basic Header contains information concerning to the size length of the structure, time and a counting records of the experiment. The Figure 2 shows the structure of the basic header and the size (in bytes) of each variable.

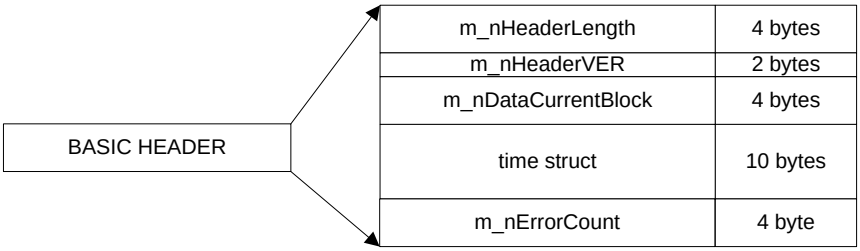

Figure 2 – Basic Header Structure (total length: 24 bytes)

The principal variables are:

a. m_nHeaderLength: total size length of the header in bytes (See Figure 3).

NOTE: For the 1st basic header, the length considers the basic header length and the long header length. For the other basic headers on the file the length is the size of the basic header (24 bytes).

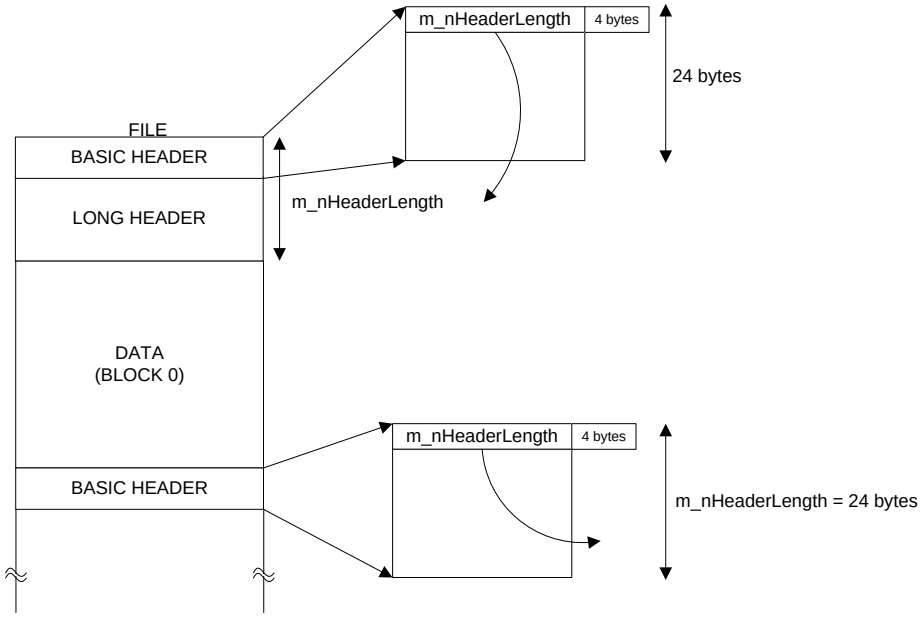

Figure 3 – variable m_nHeaderLength

b. m_nDataCurrentBlock: is a counter specifying the actual data block number on the file (e.g. Block “0”, Block “1”, …, Block “n”).

c. time struct: the start time of the data block. It is a structure described on *Figure 4.*

Time Structure	time_t time	4 bytes	Time in seconds since midnight,January 1,1970,coordinated universal time (UTC).
	mili	2 bytes	Fraction of a second in milliseconds
	short timezone	2 bytes	Difference in minutes, moving westward, between UTC and local time.
	short dstflag;	2 bytes	Nonzero if daylight savings time is currently in effect for the local time zone

Figure 4 – time structure

1.1.3 Long Header

The Long Header has the information of the experiment (e.g. IPP, Tx width pulse, Code, sampling window) and the processing parameters (e.g. Coherent integrations, decoding, etc) which are used for process the data online with these specific parameters. As shown on Figure 5, the Long Header is divided in:

- System Parameters: Contains some acquisition system variables like the number of samples to acquire. Its length is fixed to 24 bytes.
- Radar Controller Parameters: Includes most of the radar controller parameters that are used on the experiment. Some variables included on this structure are e.g. IPP [km], Tx width [km], code, etc.
- Process Parameters: Structure that contains the process parameters applied to the data online. Some of the most important are the size of each data (char, short or float), number of coherent integrations performed, decoding, etc.

Each structure has its length at the beginning as shown on Figure 5.

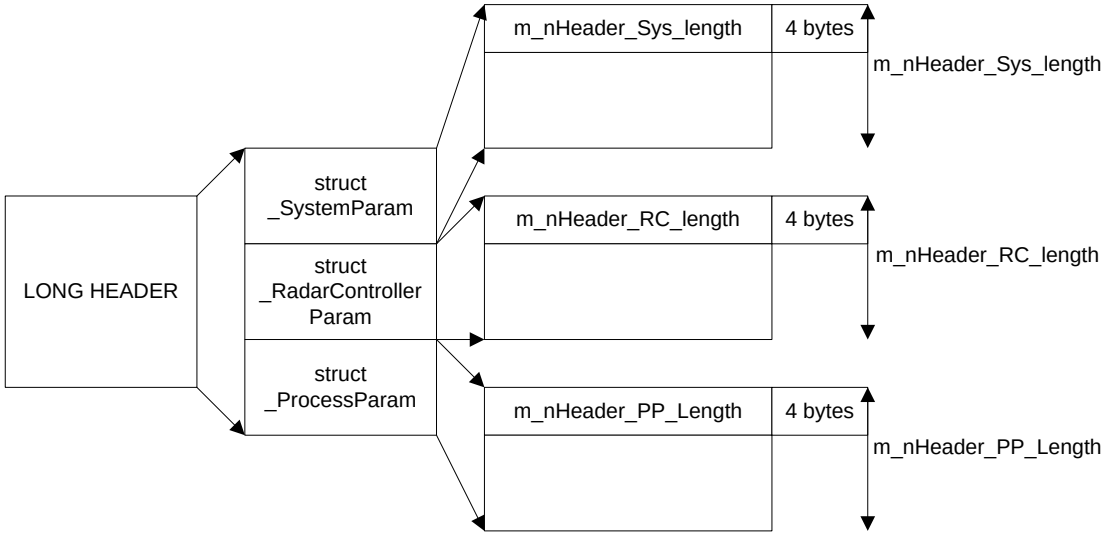

Figure 5 – Structure of Long header

A better explanation of the main variables for each structure is described below.

color	Meaning
	Length of structure
	Main variables

IMPORTANT: All the variables are saved using a 32-bit OS (Windows XP).

1.1.3.1 System Parameters

As mentioned before this structure has the acquired system variables. The main variables are:

a. m_nHeader_Sys_length: the length of the structure. Its value is fixed to 24bytes.

b. m_nSamples: number of samples acquired on each profile (ipp).

c. m_nProfiles: number of profiles (ipp's) acquired for each block of data

d. m_nChannels: number of channels acquired.

System Parameters (struct _SystemParam gRadarP)

RawData			Bytes	Description
0	unsigned long	m_nHeader_Sys_length	4	Structure Length in bytes = 24 bytes
4	unsigned long	m_nSamples	4	Number of Samples acquired per Profile
8	unsigned long	m_nProfiles	4	Number of Acquired Profiles per Block of data
12	unsigned long	m_nChannels	4	Number of Channels
16	unsigned long	m_nADCResolution	4	ADC Resolution (8 bits, 16 bits)
20	unsigned long	m_nPCIDIOBusWidth	4	PCI Card Bus Width (bits)
		TOTAL	24	

Table 1 – System parameters structure

1.1.3.2 Radar Controller Parameters

This structure has most of the radar controller variables used for the experiment. The main variables of this structure are:

a. m_nHeader_RC_length: The length of the structure may vary depending on the parameters enabled for the experiment. E.g. number of sampling windows, number of codes used

b. IPP: The time when the transmission pulse will be repeated. It is expressed in [km].

c. TxA/TxB pulse width: the width of the transmitting pulse. It is expressed in [km].

d. Parameters of Code used:

- m_nCodeType: type of code used on the experiment (e.g. Barker3, Complementary Code 4, etc). Its value is described on Appendix B (Table of Flags)

- m_nNum_Codes: number of codes. It depends on the type of code used. e.g.:

- Barker3 (1,1,-1): # of codes is 1
- Complementary Code 4 ((1,1,1,-1);(1,1,-1,1)): # of codes is 2

- m_nNum_Bauds: number of bauds on the code. It depends on the type of code used. E.g.:
  - Barker3 (1,1,-1): # of bauds is 3
  - Complementary Code 4 ((1,1,1,-1);(1,1,-1,1)): # of bauds for each code is 4
- m_snCode[m_nNum_Codes] : the code used. E.g.: ((1,1,1,-1);(1,1,-1,1))

e. Parameters of sampling window:

- m_nNum_Windows: Number of sampling windows of the experiment.
- m_sfH0[m_nNum_Windows]: The first height of the sampling window. It is expressed in [km].
- m_sfDH[m_nNum_Windows]: The resolution used on that window (resolution: how many kms are between 2 samples). It is expressed in [km].
- m_snNSA[m_nNum_Windows]: the number of samples to acquire on each window of the experiment.

On Table 2 is shown the complete radar controller structure. As it can be seen the length of the structure is fixed (116) until the sampling window declaration starts. After that the length of each variable will depend if it is used or not. So the total length of the structure will be 116 plus the length of the dynamic variables.

Radar Controller Parameters(struct _SystemParam RCparam)

#bytes	RawData		Bytes	Description
0	Unsigned long	m_nHeader_RC_length	4	Structure Length in bytes
4	Unsigned long	m_nEspType	4	Experiment type. 0 : normal; 1: join
8	Unsigned long	m_nNTX	4	Number of IPP's of the complete sequence
12	Float	m_fIPP	4	Distance between transmissions (km)
16	Float	m_fTXA	4	Pulse width on TxA (km)
20	Float	m_fTXB	4	Pulse width on TxB (km)
24	Unsigned long	m_nNum_Windows	4	Number of sampling Windows
28	Unsigned long	m_nNum_Taus	4	Number of delays from B relative to A
32	Unsigned long	m_nCodeType	4	**See Table of Flags (appendix B)**
36	Unsigned long	m_nL6_Function	4	**See Table of Flags (appendix B)**
40	Unsigned long	m_nL5_Function	4	**See Table of Flags (appendix B)**
44	Float	m_fCLOCK	4	Frequency of the clock for the pulses (1.00Mhz o 1.2 Mhz)
48	Unsigned long	m_nPrePulseBefore	4	Typical 12 (us)
52	Unsigned long	m_nPrePulseAfter	4	Typical 1 (us)
56	char	m_sRango_TR[16]	16	Range of TR transmitting pulses
72	Unsigned long	m_nDinFlags	4	**See Table of Flags (appendix B)**
76	char	m_sRango_TXA[20]	20	Range of TxA transmitting pulses

96	char	m_sRango_TXB[20]	20	Range of TxB transmitting pulses
116	Float	m_sfH0[m_nNum_Windows]	12*m_nNum_Windows	
	Float	m_sfDH[m_nNum_Windows]		
	Unsigned long	m_snNSA[m_nNum_Windows]		
	Float	m_sfTau[m_nNum_Taus]	4*m_nNum_Taus	
	Unsigned long	m_nNum_Codes	4*(m_nCodeType != NONE)	
	Unsigned long	m_nNum_Bauds	4*(m_nCodeType != NONE)	
	Unsigned long	m_snCode[m_nNum_Codes]	4*m_nNum_Codes*(m_nCodeType != NONE) * ((int)(m_nNum_Bauds/32)+1)	
	Unsigned long	m_nFLIP1	4*(m_nL5_Function==FLIP)	
	Unsigned long	m_nFLIP2	4*(m_nL6_Function==FLIP)	
	Unsigned long	m_nL5_Num_Windows	4*(m_nL5_Function==SAMPLING)	
	Float	m_sL5_fH0[m_nL5_Num_Window]	12*m_nL5_Num_Windows * (m_nL5_Function==SAMPLING)	
	Float	m_sL5_fDH[m_nL5_Num_Window]		
	Unsigned long	m_sL5_nNSA[m_nL5_Num_Windows]		
	Unsigned long	m_nL5_Num_Codes	4*(m_nL5_Function==CODE)	
	Unsigned long	m_nL5_Num_Bauds	4*(m_nL5_Function==CODE)	
	Unsigned long	m_sL5_nCode[m_nL5_Num_Codes]	4*m_nL5_Num_Codes*(m_nL5_Function==CODE) * ((int)(m_nL5_Num_Bauds/32)+1)	
	Unsigned long	m_nL6_Num_Windows	4*(m_nL6_Function==SAMPLING)	
	Float	m_sL6_fH0[m_nL6_Num_Window]	12*m_nL6_Num_Windows * (m_nL6_Function==SAMPLING)	
	Float	m_sL6_fDH[m_nL6_Num_Window]		
	Unsigned long	m_sL6_nNSA[m_nL6_Num_Windows]		
	Unsigned long	m_nL6_Num_Codes	4*(m_nL6_Function==CODE)	
	Unsigned long	m_nL6_Num_Bauds	4*(m_nL6_Function==CODE)	
	Unsigned long	m_sL6_nCode[m_nL6_Num_Codes]	4*m_nL6_Num_Codes*(m_nL6_Function==CODE) * ((int)(m_nL6_Num_Bauds/32)+1)	
	Unsigned long	m_nSynchro_Delay	4*(m_nDinFlags&FLAG_RC_SYNC_DELAY_ESP)	
	Unsigned long	m_nExt_Synchro_Divisor	4*(m_nDinFlags&FLAG_RC_SYNC_DIV_ESP)	
	Unsigned long	m_nExt_Clk_Divisor	4*(m_nDinFlags&FLAG_RC_CLK_DIV_ESP)	
	Unsigned long	m_nExt_Synchro_Delay	4*(m_nDinFlags & FLAG_RC_EXT_SYNC_DELAY_ESP)	
	Unsigned long	m_nTR_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TR_DYNAMIC)	
	char	m_sTR_Range[m_nTR_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TR_DYNAMIC) * (m_nTR_RangeLen+1)	
	Unsigned long	m_nTXA_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TXA_DYNAMIC)	
	char	m_sTXA_Range[m_nTXA_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TXA_DYNAMIC) * (m_nTXA_RangeLen+1)	
	Unsigned long	m_nTXB_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TXB_DYNAMIC)	
	char	m_sTXB_Range[m_nTXB_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TXB_DYNAMIC) * (m_nTXB_RangeLen+1)	
			+12*m_nNum_Windows +4*m_nNum_Taus +(8+4*m_nNum_Codes)*(m_nCodeType!=NONE) +4*(m_nL5_Function==FLIP) +4*(m_nL6_Function==FLIP)	

Table 2 – Radar Controller structure

1.1.3.3 Process Parameters

This structure has the process parameters applied to the acquired data online. The main variables are:

a. m_nHeader_PP_Length : The length of the structure may vary depending on the process parameters enabled for the experiment. E.g. defined process codes, number of process windows, etc.

b. m_nDataType: Indicates if the file is RawData (value=0) or Spectra (value=1)

c. m_nSizeOfDataBlock: size of the block of data in bytes.

d. m_nProfilesperBlock: number of profiles (ipp's) contained in a block of data for Raw Data. FFT points for Spectra data.

e. Parameters of Process Windows

   This has similar variables as the sampling window. The values are obtained automatically after the processes performed to the data.

   NOTE: If not processing (decoding, filter in height) is done the values are the same as in "sampling window".

   IMPORTANT: These are the final values for the sampling window.

   - m_nData_Windows: number of process windows
   - m_sfH0[m_nNum_Windows]: first height of the process window. It is expressed in [km].
   - m_sfDH[m_nNum_Windows]: resolution of the process window. It is expressed in [km].
   - m_snNSA[m_nNum_Windows]: number of samples of the process window. It is expressed in [km].

f. m_nProcessFlags: most of the process information is contained on this variable. Some of them are:

   - Flag of coherent integrations done online
   - Flag of decode the data online
   - Flag of spectra calculation
   - Flag of incoherent integration done online
   - Size of the data (char, short or Float)
   - arrange of the data (Contiguous channels, Contiguous heights or Contiguous profiles)
   - acquisition system used for the experiment

For more information go to the appendix B (Table of Flags)

E.g. If m_nProcessFlags has the value of 0x00081403 then the experiment has the following processes:

PROCESSES	**VALUE OF FLAG**
Coherent Integration	(0x00000001)
Decode	(0x00000002)
Size of data is FLOAT	(0x00000400)
Data arranged in contiguous channels	(0x00001000)
Acquisition system used is Echotek	(0x00080000)
ProcessFlags	**0x00081403**

g. m_nCoherentIntegrations: number of coherent integrations done to the data online.

h. Process Codes: If the user wants to decode with a different code that the one used to transmit, then this variables are used.

- m_nProcessCodes: number of codes
- m_nProcessBauds: number of bauds of each code
- m_sfProcessCode[][]: code used

Process Parameters: struct _ProcessParam ProcessParam

			Bytes	Description
0	unsigned long	m_nHeader_PP_Length;	4	Structure Length in bytes
4	unsigned long	m_nDataType	4	Data type on the file :{RAWDATA = 0 ; SPECTRA =1}
8	unsigned long	m_nSizeOfDataBlock	4	Size of Data Block
12	unsigned long	m_nProfilesperBlock	4	Number of profiles per Block(RawData) or FFT Points (SpectraData)
16	unsigned long	m_nDataBlocksperFile	4	Number of Blocks per File
20	unsigned long	m_nData_Windows	4	Number of data windows
24	unsigned long	m_nProcessFlags	4	**See Table of Flags**
28	unsigned long	m_nCoherentIntegrations	4	Coherent Integrations
32	unsigned long	m_nIncoherentIntegrations	4	Incoherent Integrations
36	unsigned long	m_nTotalSpectra	4	Number of Spectra Combinations

40	Float	m_sfH0[m_nNum_Windows]	12*m_nData_Windows
	Float	m_sfDH[m_nNum_Windows]	
	unsigned long	m_snNSA[m_nNum_Windows]	
	unsigned char	m_nSpectraCombinations	2*m_nTotalSpectra
	unsigned long	m_nProcessCodes	4*(FLAG_DEFINE_PROCESS_CODE)
	unsigned long	m_nProcessBauds	4*(FLAG_DEFINE_PROCESS_CODE)
	Float	m_sfProcessCode[][]	4*m_nProcessCodes*m_nProcessBauds *(FLAG_DEFINE_PROCESS_CODE)
	unsigned long	m_nExp_NameLen	4*(m_nProcessFlags&FLAG_EXP_NAME_ESP)
	char	m_sExp_Name[m_nExp_NameLen+1]	(m_nProcessFlags&FLAG_EXP_NAME_ESP) * (m_nExp_NameLen+1)

Table 3 – Process parameters structure

1.1.4 Data Block

- Raw Data Block:

  Any raw data block on the file will have the structure shown on Figure 6.

  Using the following parameters:

  - "i" channels (*m_nChannels* - system parameter structure),
  - "h" heights (*m_snNSA* - process parameters)
  - "m" profiles per block (*m_nProfilesperBlock* - process parameter structure)
  - "T" blocks per file (m_nDataBlocksperFile - process parameter structure)

- Process Data Block:

  Any process data block on the file will usually* have the structure shown on Figure 7.

  Using the following parameters:

  - "i" channels (*m_nChannels* - system parameter structure),
  - "h" heights (*m_snNSA* - process parameters)
  - "fft" FFT points used (*m_nProfilesperBlock* - process parameter structure)
  - "T" blocks per file (m_nDataBlocksperFile - process parameter structure)

  *It depends on the configuration, e.g. no Cross Spectra pairs, no SelfSpectra pairs, and so on.

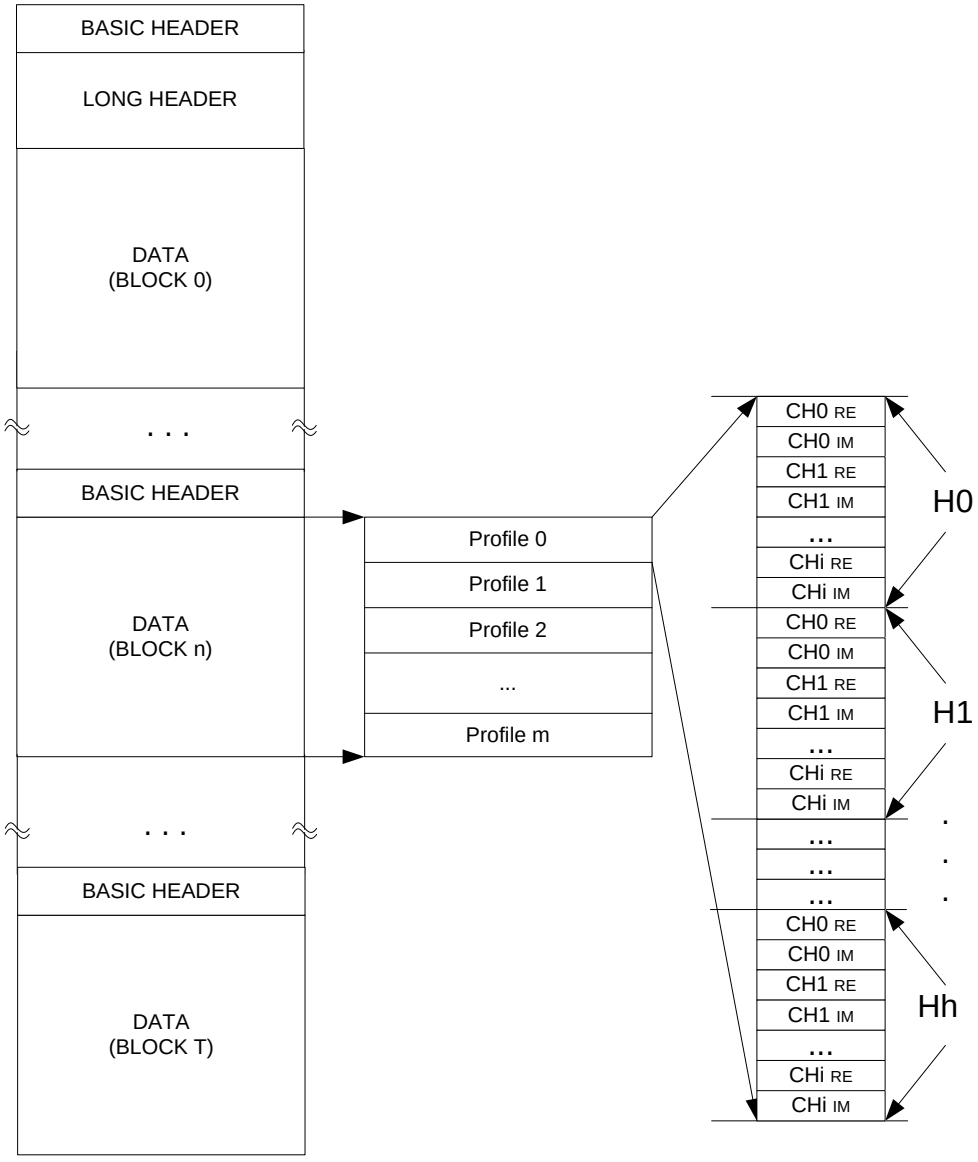

Figure 6 – File structure, Data block structure and Profile structure

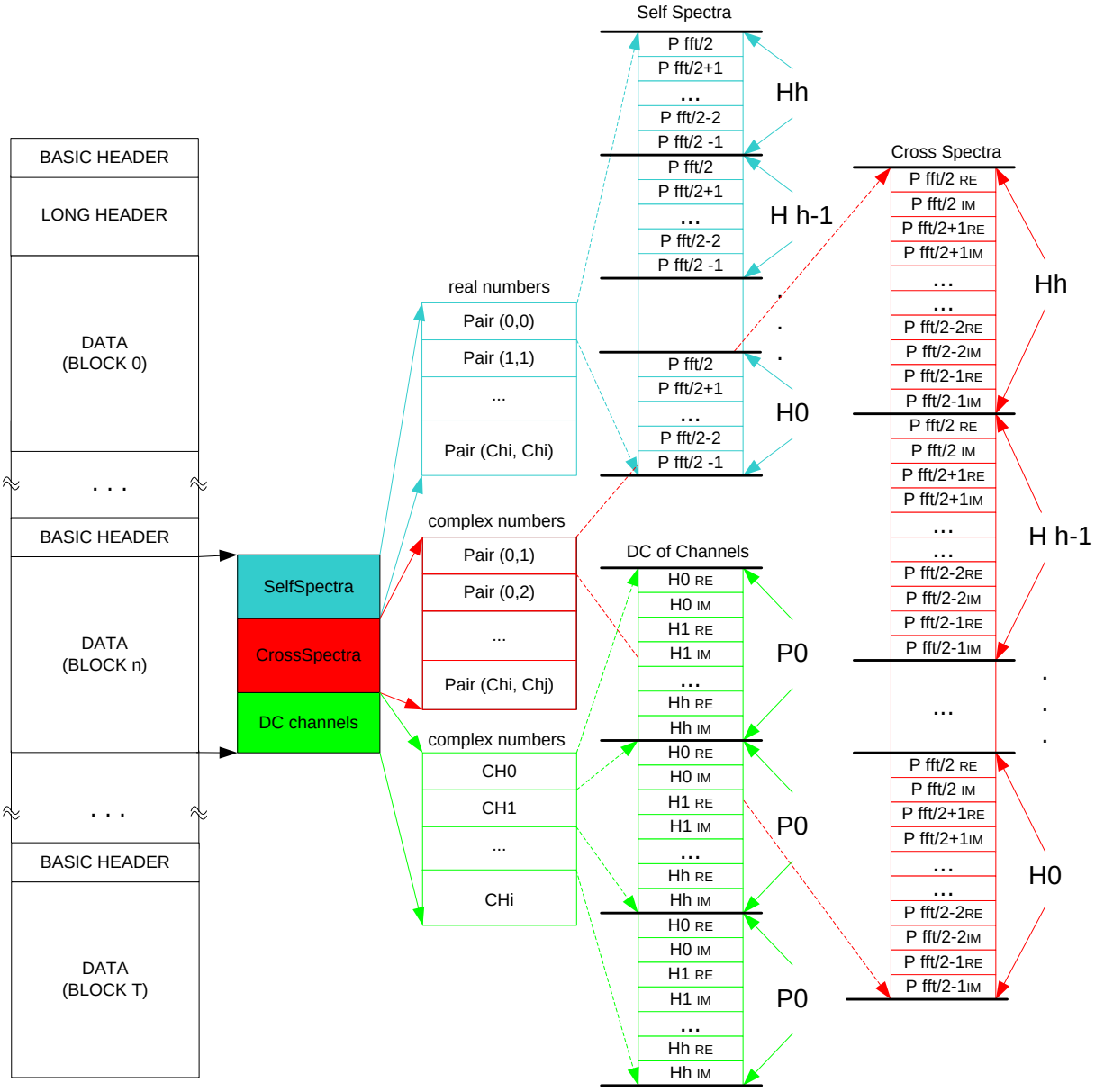

Figure 7- File structure, Data block structure, Self Spectra, Cross Spectra and DC structure

Appendix A

		Versión 1.1.0.3		
	Every block of data begins with the following basic Header structure			
Ind			Bytes	Description
0	unsigned long	m_nHeaderLength	4	Header length in bytes (First header has a dynamic length, the other headers have static length of 24 bytes)
4	unsigned short	m_nHeaderVER	2	Header Version = 1103
6	unsigned long	m_nDataCurrentBlock	4	Actual Block Number
10	_timeb	Tstruct	10	Acquired Time Structure (Begin of the data block)
20	unsigned long	m_nErrorCount_IncoInteg	4	Error counter
			24	
	Time Structure			
		Struct _timeb {	Bytes	
		time_t time;	4	Time in seconds since midnight,January 1,1970,coordinated universal time (UTC).
		unsigned short millitm;	2	Fraction of a second in milliseconds
		short timezone;	2	Difference in minutes, moving westward, between UTC and local time.
		short dstflag;	2	Nonzero if daylight savings time is currently in effect for the local time zone
		};		
			10	
For the first header in a file the following structures come after the basic header structure of 24 bytes				
			Bytes	Descripción
24	struct _SystemParam	gRadarP	24	System Parameters

48	struct _RadarControllerParam	gRControllerP	116+12*m_nNum_Windows+4*m_nNum_Taus+ (8+4*m_nNum_codes)(m_nCodeType<>0)
	struct _ProcessParam	gProcessP	40+2*m_nTotalSpectra+12*m_nData_Windows

Observation: The Length of each structure (gRControllerP, gProcessP) is written in the first parameter of them

	System Parameters	(struct _SystemParam gRadarP)		
	RawData		Bytes	Description
0	unsigned long	m_nHeader_Sys_length	4	Structure Length in bytes = 24 bytes
20	unsigned long	m_nSamples	4	Number of Samples per Profile
24	unsigned long	m_nProfiles	4	Number of Acq. Profiles per Block of data
28	unsigned long	m_nChannels	4	Number of Channels
32	unsigned long	m_nADCResolution	4	ADC Resolution (8 bits, 16 bits)
36	unsigned long	m_nPCIDIOBusWidth	4	PCI Card Bus Width (bits)
			24	

	Radar Controller Parameters(struct _SystemParam RCparam)			
	RawData		Bytes	Description
0	unsigned long	m_nHeader_RC_length	4	Experiment type. 0 : normal; 1: join
4	unsigned long	m_nEspType	4	Number of IPP's of the complete sequence
8	unsigned long	m_nNTX	4	Distance between transmissions (km)
12	float	m_fIPP	4	Pulse width on TxA (km)
16	float	m_fTXA	4	Pulse width on TxB (km)
20	float	m_fTXB	4	Number of sampling Windows
24	unsigned long	m_nNum_Windows	4	Number of delays from B relative to A
28	unsigned long	m_nNum_Taus	4	**See Table of Flags**
32	unsigned long	m_nCodeType	4	**See Table of Flags**
36	unsigned long	m_nL6_Function	4	**See Table of Flags**
40	unsigned long	m_nL5_Function	4	Frequency of the clock for the pulses (1.00Mhz o 1.2 Mhz)

44	float	m_fCLOCK	4	Typical 12 (us)
48	unsigned long	m_nPrePulseBefore	4	Typical 1 (us)
52	unsigned long	m_nPrePulseAfter	4	Range of TR transmitting pulses
56	char	m_sRango_TR[16]	16	**See Table of Flags**
72	unsigned long	m_nDinFlags	4	Range of TxA transmitting pulses
76	char	m_sRango_TXA[20]	20	Range of TxB transmitting pulses
96	char	m_sRango_TXB[20]	20	Experiment type. 0 : normal; 1: join
116	float	m_sfH0[m_nNum_Windows]	12*m_nNum_Windows	
	float	m_sfDH[m_nNum_Windows]		
	unsigned long	m_snNSA[m_nNum_Windows]		
	float	m_sfTau[m_nNum_Taus]	4*m_nNum_Taus	
	unsigned long	m_nNum_Codes	4*(m_nCodeType != NONE)	
	unsigned long	m_nNum_Bauds	4*(m_nCodeType != NONE)	
	unsigned long	m_snCode[m_nNum_Codes]	4*m_nNum_Codes*(m_nCodeType != NONE) * ((int)(m_nNum_Bauds/32)+1)	
	unsigned long	m_nFLIP1	4*(m_nL5_Function==FLIP)	
	unsigned long	m_nFLIP2	4*(m_nL6_Function==FLIP)	
	unsigned long	m_nL5_Num_Windows	4*(m_nL5_Function==SAMPLING)	
	float	m_sL5_fH0[m_nL5_Num_Window]	12*m_nL5_Num_Windows * (m_nL5_Function==SAMPLING)	
	float	m_sL5_fDH[m_nL5_Num_Window]		
	unsigned long	m_sL5_nNSA[m_nL5_Num_Windows]		
	unsigned long	m_nL5_Num_Codes	4*(m_nL5_Function==CODE)	
	unsigned long	m_nL5_Num_Bauds	4*(m_nL5_Function==CODE)	
	unsigned long	m_sL5_nCode[m_nL5_Num_Codes]	4*m_nL5_Num_Codes*(m_nL5_Function==CODE) * ((int)(m_nL5_Num_Bauds/32)+1)	
	unsigned long	m_nL6_Num_Windows	4*(m_nL6_Function==SAMPLING)	
	float	m_sL6_fH0[m_nL6_Num_Window]	12*m_nL6_Num_Windows * (m_nL6_Function==SAMPLING)	
	float	m_sL6_fDH[m_nL6_Num_Window]		
	unsigned long	m_sL6_nNSA[m_nL6_Num_Windows]		
	unsigned long	m_nL6_Num_Codes	4*(m_nL6_Function==CODE)	

	unsigned long	m_nL6_Num_Bauds	4*(m_nL6_Function==CODE)
	unsigned long	m_sL6_nCode[m_nL6_Num_Codes]	4*m_nL6_Num_Codes*(m_nL6_Function==CODE) * ((int)(m_nL6_Num_Bauds/32)+1)
	unsigned long	m_nSynchro_Delay	4*(m_nDinFlags&FLAG_RC_SYNC_DELAY_ESP)
	unsigned long	m_nExt_Synchro_Divisor	4*(m_nDinFlags&FLAG_RC_SYNC_DIV_ESP)
	unsigned long	m_nExt_Clk_Divisor	4*(m_nDinFlags&FLAG_RC_CLK_DIV_ESP)
	unsigned long	m_nExt_Synchro_Delay	4*(m_nDinFlags & FLAG_RC_EXT_SYNC_DELAY_ESP)
	unsigned long	m_nTR_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TR_DYNAMIC)
	char	m_sTR_Range[m_nTR_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TR_DYNAMIC) * (m_nTR_RangeLen+1)
	unsigned long	m_nTXA_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TXA_DYNAMIC)
	char	m_sTXA_Range[m_nTXA_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TXA_DYNAMIC) * (m_nTXA_RangeLen+1)
	unsigned long	m_nTXB_RangeLen	4*(m_nDinFlags&FLAG_RC_RANGE_TXB_DYNAMIC)
	char	m_sTXB_Range[m_nTXB_RangeLen+1]	(m_nDinFlags&FLAG_RC_RANGE_TXB_DYNAMIC) * (m_nTXB_RangeLen+1)
			+12*m_nNum_Windows +4*m_nNum_Taus +(8+4*m_nNum_Codes)*(m_nCodeType!=NONE) +4*(m_nL5_Function==FLIP) +4*(m_nL6_Function==FLIP)

	Process Parameters:	struct _ProcessParam ProcessParam		
			Bytes	Description
0	unsigned long	m_nHeader_PP_Length;	4	Structure Length in bytes
4	unsigned long	m_nDataType	4	Data type on the file :{RAWDATA = 0 ; SPECTRA =1}
8	unsigned long	m_nSizeOfDataBlock	4	Size of Data Block
12	unsigned long	m_nProfilesperBlock	4	Number of profiles per Block(RawData) or FFT Points (Spectra)

16	unsigned long	m_nDataBlocksperFile	4	Number of Blocks per File
20	unsigned long	m_nData_Windows	4	Number of data windows
24	unsigned long	m_nProcessFlags	4	**See Table of Flags**
28	unsigned long	m_nCoherentIntegrations	4	Coherent Integrations
32	unsigned long	m_nIncoherentIntegrations	4	Incoherent Integrations
36	unsigned long	m_nTotalSpectra	4	Number of Spectra Combinations
40	float	m_sfH0[m_nNum_Windows]	12*m_nData_Windows	
	float	m_sfDH[m_nNum_Windows]		
	unsigned long	m_snNSA[m_nNum_Windows]		
	unsigned char	m_nSpectraCombinations	2*m_nTotalSpectra	
	unsigned long	m_nProcessCodes	4*(FLAG_DEFINE_PROCESS_CODE)	
	unsigned long	m_nProcessBauds	4*(FLAG_DEFINE_PROCESS_CODE)	
	float	m_sfProcessCode[][]	4*m_nProcessCodes*m_nProcessBauds * (FLAG_DEFINE_PROCESS_CODE)	
	unsigned long	m_nExp_NameLen	4*(m_nProcessFlags&FLAG_EXP_NAME_ESP)	
	char	m_sExp_Name[m_nExp_NameLen+1]	(m_nProcessFlags&FLAG_EXP_NAME_ESP) * (m_nExp_NameLen+1)	

Appendix B

Table of Flags

```
/********ProcessParam.m_nDataType ************/
#define RAWDATA                                 =0x00000000
#define SPECTRA                                 =0x00000001
#define 2DIFF_PROCESS_WIN                       =0x00001000
#define 3DIFF_PROCESS_WIN                       =0x00010000
#define 4DIFF_PROCESS_WIN                       =0x00011000
#define SAVE_INCOH_INT_TIME_AVER                =0x00100000
/****Rcparam.(m_nL5_Function & m_nL6_Function********/
#define NONE                                    = 0
#define FLIP                                    = 1
#define CODE                                    = 2
#define SAMPLING                                = 3
#define LIN6DIV256                              = 4
#define SYNCHRO                                 = 5
/********RCparam.m_nCodeType       ************/
#define CODE_NONE                               = 0
#define CODE_USERDEFINE                         = 1
#define CODE_BARKER2                            = 2
#define CODE_BARKER3                            = 3
#define CODE_BARKER4                            = 4
#define CODE_BARKER5                            = 5
#define CODE_BARKER7                            = 6
#define CODE_BARKER11                           = 7
#define CODE_BARKER13                           = 8
#define CODE_AC128                              = 9
#define CODE_COMPLEMENTARYCODE2                 = 10
#define CODE_COMPLEMENTARYCODE4                 = 11
#define CODE_COMPLEMENTARYCODE8                 = 12
```

```
#define CODE_COMPLEMENTARYCODE16                = 13
#define CODE_COMPLEMENTARYCODE32                = 14
#define CODE_COMPLEMENTARYCODE64                = 15
#define CODE_COMPLEMENTARYCODE128               = 16
#define CODE_BINARY28                           = 17
/********ProcessParam.m_nProcessFlags ************/
#define FLAG_COHERENT_INTEGRATION               = 0x00000001
#define FLAG_DECODE_DATA                        = 0x00000002
#define FLAG_SPECTRA_CALC                       = 0x00000004
#define FLAG_INCOHERENT_INTEGRATION             = 0x00000008
#define FLAG_POST_COHERENT_INTEGRATION          = 0x00000010
#define FLAG_SHIFT_FFT_DATA                     = 0x00000020
#define FLAG_DATATYPE_CHAR                      = 0x00000040
#define FLAG_DATATYPE_SHORT                     = 0x00000080
#define FLAG_DATATYPE_LONG                      = 0x00000100
#define FLAG_DATATYPE_INT64                     = 0x00000200
#define FLAG_DATATYPE_FLOAT                     = 0x00000400
#define FLAG_DATATYPE_DOUBLE                    = 0x00000800
#define FLAG_DATAARRANGE_CONTIGUOUS_CH          = 0x00001000
#define FLAG_DATAARRANGE_CONTIGUOUS_H           = 0x00002000
#define FLAG_DATAARRANGE_CONTIGUOUS_P           = 0x00004000
#define FLAG_SAVE_CHANNELS_DC                   = 0x00008000
#define FLAG_DEFLIP_DATA                        = 0x00010000    =*00001*
#define FLAG_DEFINE_PROCESS_CODE                = 0x00020000    =*00010*
#define FLAG_ACQ_SYS_MASK                       = 0x001C0000    =*11100*
#define FLAG_ACQ_SYS_NATALIA                    = 0x00040000    =*00100*
#define FLAG_ACQ_SYS_ECHOTEK                    = 0x00080000    =*01000*
#define FLAG_ACQ_SYS_ADRXD                      = 0x000C0000    =*01100*
#define FLAG_ACQ_SYS_JULIA                      = 0x00100000    =*10000*
#define FLAG_ACQ_SYS_XXXXXX                     = 0x00140000    =*10100*
#define FLAG_EXP_NAME_ESP                       = 0x00200000
```

#define FLAG_CHANNEL_NAMES_ESP	= 0x00400000		
/********RadarController.m_nDinFlags ************/			
#define FLAG_RC_L4_TXA_REF	= 0x00000001	xx00000001	
#define FLAG_RC_L4_TXB_REF	= 0x00000002	xx00000010	2 bits
#define FLAG_RC_L4_TXX_REF	= 0x00000003	xx00000011	
#define FLAG_RC_L5_TXA_REF	= 0x00000004	xx00000100	
#define FLAG_RC_L5_TXB_REF	= 0x00000008	xx00001000	2 bits
#define FLAG_RC_L5_TXX_REF	= 0x0000000C	xx00001100	
#define FLAG_RC_L6_TXA_REF	= 0x00000010	xx00010000	
#define FLAG_RC_L6_TXB_REF	= 0x00000020	xx00100000	2 bits
#define FLAG_RC_L6_TXX_REF	= 0x00000030	xx00110000	
#define FLAG_RC_L7_TXA_REF	= 0x00000040	xx01000000	
#define FLAG_RC_L7_TXB_REF	= 0x00000080	xx10000000	2 bits
#define FLAG_RC_L7_TXX_REF	= 0x000000C0	xx11000000	
#define FLAG_RC_RESERVED_REF	= 0x00000000	0000000000	
#define FLAG_RC_MIDDLE_OF_SUB_BAUD_REF	= 0x00000100	0100000000	2 bits
#define FLAG_RC_MIDDLE_OF_TX_REF	= 0x00000200	1000000000	
#define FLAG_RC_BEGIN_OF_TX_REF	= 0x00000300	1100000000	
#define FLAG_RC_SYNC_DELAY_ESP	= 0x00000400		
#define FLAG_RC_PULSE_AFTER_WINDOW	= 0x00000800		
#define FLAG_RC_CTRL_SWITCH1	= 0x00001000		
#define FLAG_RC_CTRL_SWITCH2	= 0x00002000		
#define FLAG_RC_CLK_DIV_ESP	= 0x00004000		
#define FLAG_RC_RANGE_TR_DYNAMIC	= 0x00008000		
#define FLAG_RC_RANGE_TXA_DYNAMIC	= 0x00010000		
#define FLAG_RC_RANGE_TXB_DYNAMIC	= 0x00020000		
#define FLAG_RC_SYNC_DIV_ESP	= 0x00040000		
#define FLAG_RC_EXT_SYNC_DELAY_ESP	= 0x00080000		